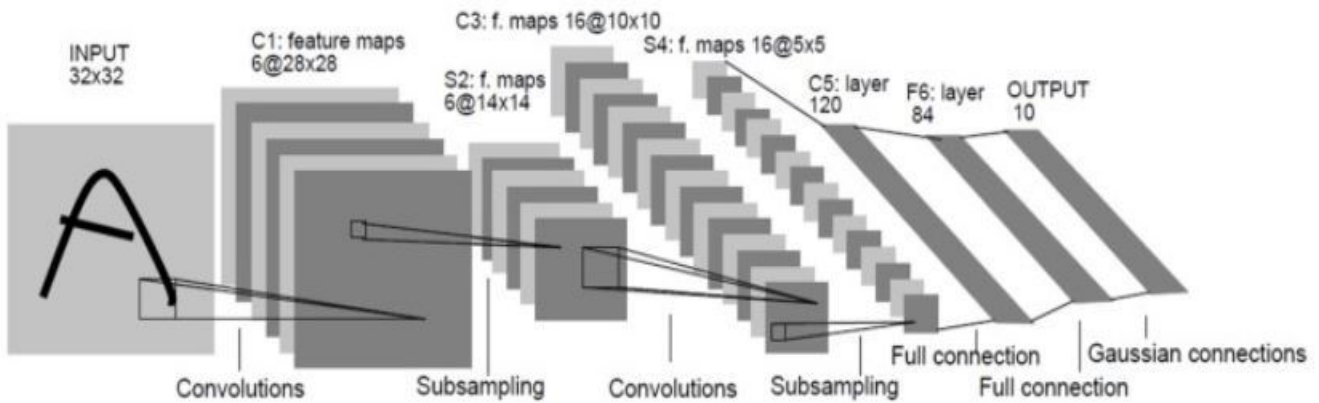


HW5 MNIST Assignment
 데이터분석과 통계의 이해, 장한휘 교수님
 전자공학과 201820885 김준성



MNIST데이터를 분석하기위한 CNN은 LeNet-5 Network를 사용하였다. 이미지 분류용 CNN중에서 조상적인 Network이다. 위 그림에서 볼 수 있듯이 LeNet-5는 Input, 3개의 Convolution layer(C1,C3,C5) 2개의 Subsampling layer(S2, S4), 1층의 Full-connected layer(F6), Output layer로 구성되어있다. Activation Function으로는 ReLU Function을 사용했다.

- 1) C1 레이어: 입력 영상(32 x 32 사이즈의 이미지)을 6 개의 5 x 5 필터로 컨볼루션 연산을 해준다. 그 결과 6 장의 28 x 28 feature maps 을 얻게 된다.
- 2) S2 레이어: 6 장의 28 x 28 feature maps 에 대해 서브샘플링을 진행한다. 결과적으로 28 x 28 사이즈의 feature maps 이 14 x 14 로 감소된다. 2 x 2 필터를 stride 2 로 설정해서 서브샘플링해주기 때문이다. 사용하는 서브샘플링 방법은 평균 풀링(average pooling)이다.
- 3) C3 레이어: 6 장의 14 x 14 특성맵에 컨볼루션 연산을 수행해서 16 장의 10 x 10 특성맵을 산출해낸다.
- 4) S4 레이어: 16 장의 10 x 10 특성 맵에 대해서 서브샘플링을 진행해 16 장의 5 x 5 특성 맵으로 축소시킨다.
- 5) C5 레이어: 16 장의 5 x 5 특성맵을 120 개 5 x 5 x 16 사이즈의 필터와 컨볼루션 해준다. 결과적으로 120 개 1 x 1 특성맵이 산출된다.
- 6) F6 레이어: 84 개의 유닛을 가진 피드포워드 신경망이다. C5 의 결과를 84 개의 유닛에 연결시킨다.
- 7) 아웃풋 레이어: 10 개의 Euclidean radial basis function(RBF) 유닛들로 구성되어있다. 각각 F6 의 84 개 유닛으로부터 인풋을 받는다. 최종적으로 이미지가 속한 클래스를 알려준다.

구현은 LeNet-5 을 모델링하였다. 현재 사용하는 맥북이 Pytorch 의 Cuda 를 지원하지 않고, 외장 그래픽카드가 존재하지않아 학습시키는데 시간이 꽤 걸렸다. 일단 결과를 확인하기 위해 반복횟수를 줄였다. 코드에는 정상적인 수치가 기입되어있다(n_epochs) 학습을 시킨 후, Train data set 의 label 을 이용하여 터미널 상에서 테스트할 사진들을 생성해 준 뒤, 터미널에서 메인함수와 함께 사진을 입력으로 넣어주어 값을 추측하였다.

```
~/Doc/G/python_assignments/assignment5-MNIST | master !2 73 | python3 main.py one.png
Epoch1, Training loss2.285015106201172
Epoch10, Training loss0.8409702181816101
Epoch20, Training loss0.5189012289047241
Epoch30, Training loss0.7012192606925964
Epoch40, Training loss0.5791622996330261
Epoch50, Training loss0.3611706793308258
Epoch60, Training loss0.15763939917087555
Epoch70, Training loss0.3652857542037964
Epoch80, Training loss0.28945624828338623
Epoch90, Training loss0.5824853777885437
Epoch100, Training loss0.14395871758460999
Epoch110, Training loss0.5760079622268677
Epoch120, Training loss0.5756868720054626
Epoch130, Training loss0.28931376338005066
Epoch140, Training loss0.36028754711151123
Epoch150, Training loss0.21586747467517853
Assume : 1
```

1. Layer C1: Convolution Layer (num_kernels=6, kernel_size=5x5, padding=0, stride=1)
2. Layer S2: Average Pooling Layer (kernel_size=2x2, padding=0, stride=2)
3. Layer C3: Convolution Layer (num_kernels=16, kernel_size=5x5, padding=0, stride=1)
4. Layer S4: Average Pooling Layer (kernel_size=2x2, padding=0, stride=2)
5. Layer F5: Fully Connected Layer (out_features=140)
6. Layer F6: Fully Connected Layer (out_features=84)
7. Layer F7: Fully Connected Layer (out_features=10) [2]

[참고문헌]

<https://m.blog.naver.com/PostView.nhn?blogId=laonple&logNo=220648539191&proxyReferer=https:%2F%2Fwww.google.com%2F>